

IMAGE GENERATION METHOD AND DEVICE USED THEREFOR

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] The present application claims priority from Japanese Patent Application Nos. 2001-25376 filed on February 1, 2001, and 2001-254576 filed on August 24, 2001, the disclosures of which are hereby incorporated by reference herein.

BACKGROUND OF THE INVENTION

[0002] The present invention relates to an image generation method and a device used therefor for generating three-dimensional images to be drawn on a two-dimensional screen, such as on a television monitor; an image processing program for making a computer execute image generation processing; and a recording medium having recorded thereon such image processing program..

[0003] There are accelerating trends in higher integration and faster processing speeds of processors and memories in recent television game machines and personal computers which enable real-time generation of three-dimensional images with real presence and perspective in the display thereof on two-dimensional monitor screens.

[0004] When a three-dimensional image is drawn on a two-dimensional monitor screen, the three-dimensional polygon data are subjected to various geometric processing,

such as coordinate conversion, clipping and lighting, and the resultant data are further subjected to translucent projection conversion.

[0005] In the conventional image generation device for generating a three-dimensional image to be displayed on a two-dimensional monitor screen, drawing of a semi-transparent object or drawing based on pixel occupational factors for anti-aliasing of the edge portion of an object generally requires so-called α -blending, which is a technique of generating a drawn image based on linear interpolation between pixel values on a frame buffer and drawn pixel values using a coefficient α for expressing semi-transparency (or transparency).

[0006] It has, however, been known for such technique of generating a drawn image based on α -blending that an image will look unnatural unless the objects are drawn in a manner such that the further the object is in the depth direction of the screen (direction of the depth from the viewpoint, referred to as Z direction hereinafter), the earlier it should be drawn. There is also known a technique for drawing objects according to their depth in the Z direction, such as a Z buffer method which can provide correct depth relation among the individual planes by finding out the closest planes for the individual pixels and expressing each pixel with a color of such closest plane. This is unlike the Z sorting method whereby the individual planes are painted with respective colors. With the Z buffer method, however, the foregoing

problem remains unsolved.

SUMMARY OF THE INVENTION

[0007] The present invention was proposed to address the foregoing problem, and an object thereof resides in providing an image generation method and a device used therefor, an image processing program for making a computer execute image generation processing, and a recording medium having recorded thereon such image processing program, all of which being aimed at successfully providing an image based on α -blending in a natural and unfailing manner without using a technique such as Z sorting.

[0008] The present invention ensures that object images to be subjected to a predetermined image processing are subjected to such image processing depending on predetermined coefficients of semi-transparency, and that a portion of the object images closest to a virtual viewpoint is processed last.

[0009] That is, the present invention is successful in providing an image based on a first image processing (α -blending, for example) depending on coefficients of semi-transparency in a natural and unfailing manner without using a technique such as Z sorting, by always allowing the portion of the object images closest to the virtual viewpoint to be processed later than any other object images to be subjected to the predetermined image processing depending on the coefficients of semi-transparency.

[0010] Other and further objects and features of the present invention will become obvious upon an understanding of the illustrative embodiments about to be described in connection with the accompanying drawings or will be indicated in the appended claims, and various advantages not referred to herein will occur to one skilled in the art upon employing the invention in practice.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] Fig. 1 is a block diagram explaining a principal portion of an image generation device according to an embodiment of the present invention;

[0012] Fig. 2 is a flow chart showing a process for generating a two-dimensional image using two frame data storage areas in the image generation device according to the present embodiment;

[0013] Fig. 3 is a schematic drawing showing an exemplary drawn image obtained in step S3 in the flow chart shown in Fig. 2;

[0014] Fig. 4 is a schematic drawing showing an exemplary drawn image obtained in step S4 in the flow chart shown in Fig. 2;

[0015] Fig. 5 is a schematic drawing showing an exemplary drawn image obtained in step S6 in the flow chart shown in Fig. 2;

[0016] Fig. 6 is a schematic drawing showing an exemplary drawn image obtained in step S7 in the flow chart shown in

Fig. 2; and

[0017] Fig. 7 is a block diagram explaining a personal computer for executing an image processing program whose process flow was shown by the flow chart in Fig. 2.

DETAILED DESCRIPTION

[0018] Various embodiments of the present invention will be described with reference to the accompanying drawings. It is to be noted that the same or similar reference numerals are applied to the same or similar parts and elements throughout the drawings, and the description of the same or similar parts and elements will be omitted or simplified.

A Configuration of an Image Generation Device of the Embodiment

[0019] Fig. 1 shows a schematic configuration of an image generation device 1 of the embodiment of the present invention. The image generation device 1 of the embodiment is a device for generating a two-dimensional image based on texture mapping onto three-dimensional polygons, and is applicable to video game machines, personal computers and three-dimensional graphic devices.

[0020] The image generation device 1 shown in Fig. 1 mainly includes a luminance calculation and coordinate conversion unit 2, an LOD (level of detail) calculation unit 3, a texture coordinate calculation unit 4, a DDA (digital differential analyzer) unit 5, a pixel engine 6 and an image memory 7.

[0021] The image memory 7 further includes a Z buffer 8

for storing values in the depth-wise direction from a viewpoint (Z coordinate values), a basic texture buffer 9 for storing basic texture data for generating overall patterns on the polygons based on mapping, a modulation-exclusive texture buffer 10 for storing texture data used for amplitude modulation of the pattern generated by the mapping of the basic texture, and two frame buffers 11, 12 for storing and synthesizing frame data (two-dimensional image data) displayed on a two-dimensional monitor screen as described later.

[0022] It is also allowable to store data for the basic texture and the modulation-exclusive texture into the frame buffer. One possible way is to store data for the basic texture into an RGB storage area in the frame buffer and data for the modulation-exclusive texture into an α storage area of such frame buffer. With such strategy, it is no longer necessary for the image memory 7 to have storage areas such as the basic texture buffer 9 and the modulation-exclusive texture buffer 10.

[0023] Typically input through an input terminal 13 of the image generation device 1 provided with such components are three-dimensional polygon information, a coefficient α for expressing semi-transparency used for α -blending, texture information, light source information and viewpoint information. This various information typically is supplied through a communication line or a storage device.

[0024] The three-dimensional polygon information

typically includes information such as (x, y, z) coordinates of the individual apexes of the triangular polygons and normal lines at such apexes; and the texture information includes information for the individual texels having values of R (red), G (green) and B (blue), which are the three primary colors of light, and texture coordinate information corresponding to the individual apexes of the triangular polygons. The coefficient α expressing semi-transparency is the one used for α -blending which defines a ratio of image blending in texture mapping, and is specifically set at values "A" ("A" represents α) corresponding to the values of R, G and B. The viewpoint information and the light source information are used for luminance calculation and coordinate conversion for the polygons. The light source information may represent not only a single light source but also a plurality of light sources. The individual apex information of the polygons can also include, besides the information described above, various information such as color information or fog values for providing an effect of fogging for distant objects.

[0025] The various information is first entered into the luminance calculation and coordinate conversion unit 2 of the image generation device 1. The luminance calculation and coordinate conversion unit 2 converts the input individual coordinate information of the polygons into coordinate values in a coordinate system for the two-dimensional drawing based on the viewpoint information, and calculates the luminance at the individual apexes of the polygons based on the viewpoint

information and the light source information. In addition to the foregoing calculations, the luminance calculation and coordinate conversion unit 2 is also responsible for other processing, such as translucent conversion. The individual values calculated in the luminance calculation and coordinate conversion unit 2 are input to the LOD calculation unit 3.

[0026] The LOD calculation unit 3 calculates, based on Z coordinate values of such converted polygons, LOD values which are used when the pixel engine 6 reads out basic textures from the basic texture buffer 9. Here the LOD value can be calculated from a shrinkage factor when a polygon is shrunk, where such shrinkage factor is typically obtained as a logarithmic value of the distance from the viewpoint to the polygon. The LOD value is sent via the texture coordinate calculation unit 4 and the DDA unit 5 to the pixel engine 6.

[0027] The texture coordinate calculation unit 4 calculates, from texture coordinate values for the basic texture, texture coordinate values which are used when the pixel engine 6 reads out modulation-exclusive texture coordinates from the modulation-exclusive texture buffer 10. These texture coordinate values are sent via the DDA unit 5 to the pixel engine 6.

[0028] The DDA unit 5 converts the two-dimensional polygon apex information, Z coordinate values, luminance information and so forth into pixel information. More specifically, the DDA unit 5 determines, by linear

interpolation, the coordinate values, Z values, luminance values and texture coordinate values for the individual pixels. Output of such DDA unit 5 is sent to the pixel engine 6.

[0029] The pixel engine 6 not only controls read-out and write-in operations to or from the Z buffer 8, the basic texture buffer 9, the modulation-exclusive texture buffer 10 and the individual frame data storage areas of the first and second frame buffers 11, 12, but is also responsible for texture mapping using both the texel information read out from the basic texture buffer 9 or the modulation-exclusive texture buffer 10, and the pixel information determined by the DDA unit 5, comparison of the Z coordinate values, calculation of the pixel values and write-in operation of the pixel values.

[0030] In such texture mapping, the pixel engine 6 performs the texture mapping onto the three-dimensional polygons based on a so-called MIP mapping system using the basic textures stored in the basic texture buffer 9. The MIP mapping system refers to a system in which textures of various scales, such as $1/2$, $1/4$, $1/8$ scales and so on (numerals represent the ratios of lengths of the individual sides), are previously registered, and a proper one of which is selected depending on a shrinkage factor to be mapped onto a three-dimensional polygon. According to such MIP mapping system, textures shrunk from the original texture are mapped onto the polygons, which successfully prevents aliasing likely to occur when the polygons are shrunk. For cases in

which the basic texture is added with still higher frequency components in such texture mapping, the pixel engine 6 performs texture mapping using the modulation-exclusive texture stored in the modulation-exclusive texture buffer 10.

[0031] The pixel engine 6 is also responsible for α -blending, scissoring, Z test, dithering, color clamping and so forth. Scissoring refers to a process for removing data which overflows the screen. The Z test refers to a process for comparing Z values of the individual pixels and Z values stored in the Z buffer 8, and drawing only pixels closer to the viewpoint while omitting distant pixels. Dithering refers to a process for staggering color pixel arrangement so as to express a large number of apparent colors with a small number of color pixels. Color clamping refers to a process for limiting calculated color values so as not to exceed 255 or become lower than 0.

[0032] The pixel engine 6 stores the image data obtained from such various processes into the first and second frame buffers 11, 12 as described later, and synthesizes frame data (two-dimensional image data) to be displayed on the two-dimensional monitor screen. Such synthesized two-dimensional image data are then read out from the first or second frame buffers 11, 12, and output through an output terminal 14 to the two-dimensional monitor device.

[illegible]

[0034] The image generation device 1 of the present embodiment has the first and second frame buffers 11, 12 (two frame data storage areas) within the image memory 7, and repeats a predetermined image drawing process two or three times while controlling ON/OFF of the α -blending as a predetermined image process, Z test as a predetermined comparative process, and a Z value write-in operation as a saving of distance information as described later, and the obtained images are synthesized based on ON/OFF control of the α -blending and Z test to thereby ensure an unfailing α -blended drawing without relying upon the Z sorting.

[0035] In other words, the pixel engine 6 of the image generation device 1 of the present embodiment can generate unfailing two-dimensional image data by repeating the image drawing process two or three times using the first and second frame buffers 11, 12 while being combined with ON/OFF controls of the α -blending, Z test and Z value write-in operation, by performing the process shown in the flow chart of Fig. 2.

11

viewpoint (Z coordinate value) increase in the order of **OB1**, **OB2**, **OB3** and **OB4**. Objects **OB1** and **OB3** are those not to be α -blended, and objects **OB2** and **OB4** are those to be α -blended.

[0037] As shown in Fig. 2, the pixel engine 6, in a first step **S1**, clears the first and second frame buffers 11, 12 in the image memory 7 (first and second frame data storage areas), and then in step **S2**, determines for each object to be arranged within a frame whether it is to be α -blended.

[0038] Next in step **S3**, the pixel engine 6 subjects the objects for which it was previously determined in step **S2** that α -blending is "not necessary", that is, the objects that are not to be α -blended, to the Z test and Z value write-in operation (Z test ON, and Z write-in operation ON), and then stores such objects into either of the two frame buffers 11 and 12. Here the first frame buffer 11 is used, for example (drawn in the frame buffer 11).

[0039] Of objects **OB1** to **OB4**, those not to be α -blended are objects **OB1** and **OB3**, where object **OB3** is more distant from the viewpoint than object **OB1**. Since the Z value write-in operation is activated (in ON state) in step **S3**, Z values of the individual pixels are written in the Z buffer 8 to draw only pixels closer to the viewpoint while not retaining the distant pixels. That is, the RGB values and the α value of the pixels closest to the viewpoint are separately written in an RGB value storage area and an α value storage area of the first frame buffer 11.

[0040] Thus, a drawn image obtained in step **S3**, in which

objects **OB1** and **OB3** are not α -blended under the conditions of the α -blending OFF, Z test ON and Z write-in operation ON, will appear as shown in Fig. 3, where the color or pattern of object **OB3** which is more distant from the viewpoint is partially hidden behind object **OB1** within a portion in which both objects overlap. Thus, such drawn image appears as a three-dimensional image in which object **OB1** is seen in front of object **OB3**.

[0041] Note that the processing in step **S3** will not be necessary when all objects in the frame are to be α -blended.

[0042] Next, the pixel engine 6 subjects the objects for which it was previously determined in step **S2** that α -blending is "necessary", that is, the objects that are to be α -blended, to the processing in step **S4** under conditions of α -blending ON, Z test ON and Z write-in operation OFF, and then overwrites such objects into the first frame buffer 11 similarly to step **S3** (drawn in the frame buffer 11).

[0043] Of objects **OB1** to **OB4**, those to be α -blended are objects **OB2** and **OB4**. Objects **OB1**, **OB2**, **OB3** and **OB4** are located more distantly from the viewpoint in this order, and the processing in step **S4** is performed under the conditions of α -blending ON, Z test ON and Z write-in operation OFF.

[0044] Thus, a drawn image obtained in step **S4**, in which objects **OB2** and **OB4** are overwritten, under the conditions of α -blending ON, Z test ON and Z write-in operation OFF, onto the image previously drawn in the first frame buffer 11 in step **S3**, will appear as shown in Fig. 4. As is known

from Fig. 4, the drawn image obtained in step **S4** is correct in terms of the depth relationship between objects **OB1** and **OB3** drawn in step **S3** under the condition of α -blending OFF, but is incorrect in terms of the depth relationship between objects **OB2** and **OB4** drawn in step **S4** under the condition of α -blending ON, which makes the drawn image unnatural. In short, the drawn image should be such that objects **OB1** to **OB4** are aligned therein so that object **OB1** comes foremost and objects **OB2**, **OB3** and **OB4** follow in this order so as to have a larger distance from the viewpoint. But in the drawn image at this point in time, the objects drawn in step **S4** under the condition of α -blending ON do not follow the correct order given by the Z coordinate values. Nevertheless, areas where objects **OB2** and **OB4** are to be drawn are assigned with approximated colors or patterns.

[0045] Next in step **S5**, the pixel engine 6 copies the Z buffer values stored in the first frame buffer 11 to the Z buffer of the second frame buffer 12.

[0046] Next in step **S6**, the pixel engine 6 processes the objects to be α -blended under the conditions of α -blending OFF, Z test ON and Z write-in operation ON, and stores such processed objects in the second frame buffer 12 (drawn in the frame buffer 12).

[0047] Of objects **OB1** to **OB4**, those to be α -blended are objects **OB2** and **OB4**, where object **OB4** is more distant than object **OB2**. In such processing in step **S6**, the conditions set for objects **OB2** and **OB4** are α -blending OFF, Z test ON

and Z write-in operation ON.

[0048] Thus, the resultant drawn image is as shown in Fig. 5, in which the color or pattern of object **OB4** which is more distant from the viewpoint is partially hidden by object **OB2** within a portion in which both objects overlap. Since the Z value write-in operation is set ON in step **S6**, the Z values of the individual pixels are written in the Z buffer **8** to draw only the pixels closer to the viewpoint while not retaining the distant pixels. That is, the RGB values and the α value of the pixels closest to the viewpoint are separately written in an RGB value storage area and an α value storage area of the second frame buffer **12**. Now the Z buffer **8** has stored therein the Z coordinate values previously used in step **S3**, which can directly be used for the Z test between the drawn image and the background image.

[0049] Next in step **S7**, the pixel engine **6** performs processing under the conditions of α -blending ON and Z test OFF for the entire image stored in the first frame buffer **11**, using the entire image stored in the second frame buffer **12** as a texture, and the obtained image is stored in the first frame buffer **11** (drawn in the first frame buffer **11**).

[0050] The drawn image, obtained in step **S7** by processing the entire image in the first frame buffer **11** under the conditions of α -blending ON and Z test OFF, using the entire image stored in the second frame buffer **12** as a texture, ensures that the portions of the objects to be α -blended which are closest to the viewpoint are always α -blended last. In

particular in α -blending, an object closest to the viewpoint can most significantly affect the quality of the image, so that drawing according to the present embodiment can provide an unfailing image.

[0051] That is, as shown in Fig. 6, the image obtained in the present embodiment is such that of objects **OB2** and **OB4** to be α -blended, object **OB2** which comes foremost is always α -blended last. As a result, the image finally obtained is such as to apparently have object **OB1** foremost, object **OB2** of those objects to be α -blended next, object **OB3** next, and object **OB4** last. The dashed line and broken line in Fig. 6 express that the color or pattern of the behind objects in the overlapped portions of the objects can be seen in a semi-transparent manner by virtue of the α -blending.

[0052] Thereafter in step **S8**, the pixel engine **6** determines whether the draw processing is necessary for the next frame data, and returns to step **S1** if necessary, and completes the processes shown in Fig. 2 if not.

[0053] In the pixel engine **6**, the processes of the flow chart shown in Fig. 2 can also be accomplished on a hardware basis such as using a DSP, or on a software basis using a processing program downloaded through a communication network or using an image generation processing program read out by a storage device from a storage medium, which are executed by a CPU. In particular, when the processing in the pixel engine **6** is accomplished on a software basis, the image generation processing program will be such as to create within

the storage area of the image memory 7 two frame data storage areas such as the foregoing first and second frame buffers 11 and 12, and then sequentially execute the processes of steps S1 to S8 previously explained referring to the flow chart of Fig. 2. Such image generation processing program may not only be provided previously as a processing program dedicated to the pixel engine 6, but also may be input together with, or prior to, the polygon information through the input terminal 13 in Fig. 1.

[0054] As one example for accomplishing the image generation processing of the present embodiment on a software basis, Fig. 7 shows a schematic configuration of a personal computer for executing the image processing program having the flow shown in Fig. 2. The image processing program of the embodiment is mainly executed by a CPU 123 shown in Fig. 7.

[0055] In Fig. 7, a storage section 128 typically comprises a hard disk and a drive therefor. Such hard disk has stored therein an operating system program, an image processing program 129 of the embodiment read out from one of various recoding media such as a CD-ROM or DVD-ROM, or downloaded through a communication line, and a variety of data 130 such as graphic information for polygon drawing, textures, Z values, general textures, color values and α values.

[0056] Communication section 121 refers to a communication device responsible for data communication with

external devices, which may be a modem for establishing connection to an analog public telephone line, a cable modem for establishing connection to a cable television network, a terminal adaptor for establishing connection to an ISDN (integrated services digital network), or a modem for establishing connection to an ADSL (asymmetric digital subscriber line). Communication interface (I/F) section 122 refers to an interface device responsible for protocol transfer for enabling transmission/reception of data between the communication section 121 and an internal bus (BUS).

[0057] An input section 133 refers to an input device, such as a keyboard, mouse or touch pad, and user interface (I/F) section 132 refers to an interface device for supplying signals from such input section 133 to the internal devices.

[0058] Drive section 135 refers to a drive device capable of reading out various data or programs including the image processing program of the embodiment from a recording medium, including a disk medium 151, such as a CD-ROM or DVD-ROM, or from a card-type or other type of semiconductor memory. Drive interface (I/F) section 134 refers to an interface device for supplying signals from the drive section 135 to the internal devices.

[0059] Display section 137 refers to a display device such as a CRT (cathode ray tube) or liquid crystal display, and display drive section 136 is a drive device for driving the display section 137.

[0060] The ROM 124 typically comprises a rewritable

non-volatile memory, such as a flash memory, which stores a BIOS (basic input/output system) and various default values. A RAM 125 will have loaded therein application programs and various data read out from a hard disk of the storage section 128, and is used as a work RAM of the CPU 123.

[0061] The CPU 123 controls the entire operation of the personal computer and executes the foregoing image generation processing based on the operating system program stored in the storage section 128 or the image processing program 129 of the present embodiment. That is, as shown in Fig. 7, the CPU 123 can accomplish the image generation processing as described in the above embodiment by executing the image processing program of the embodiment, which is one of the application programs read out from the hard disk of the storage section 128 and loaded into the RAM 125.

Summary of the Embodiment

[0062] In the present invention, object images to be subjected to predetermined image processing are subjected to such image processing and further to a predetermined comparison processing depending on assigned distances from a virtual viewpoint, and the compared object images are then drawn in a first frame image storage area without storing post-comparative assigned distances; the same object images are separately subjected to the predetermined comparison processing without subjecting them to the predetermined image processing. The compared object images are then drawn in a

second frame image storage area while storing the post-comparative assigned distances. The frame images stored in the second frame image storage area are subjected to the predetermined image processing without being followed by the predetermined comparison processing, and such frame images are then synthesized with the frame images previously stored in the first frame image storage area, thereby ensuring that the portion of the object images closest to the virtual viewpoint is always subjected to the predetermined image processing last. The embodiment of the present invention also allows the processing to be simplified if there are no objects not to be α -blended.

[0063] More specifically, according to the embodiment of the present invention, the first and second frame buffers 11 and 12 are provided; and the pixel engine 6 executes, using such buffers, a series of processes in which the objects to be α -blended are processed under the conditions of α -blending ON, Z test ON and Z value write-in operation OFF, and the resultant objects are stored typically in the first frame buffer 11; the same objects are separately processed under the conditions of α -blending OFF, Z test ON and Z value write-in operation ON, and the resultant objects are stored in the second frame buffer; and the entire image stored in the second frame buffer 12 is drawn as a texture onto the entire image stored in the first frame buffer 11 under the conditions of α -blending ON and Z test OFF. This ensures that, of all objects to be α -blended, the foremost one which can

most largely affect the image quality will always be α -blended last, which successfully provides an unfailing α -blended image without using the Z sorting.

[0064] According to the embodiment, unfailing drawing in which the correct depth relation of the objects is retained can be obtained, even if some of these objects are not involved in α -blending, which is accomplished by processing such not-involved objects under the conditions of Z test ON and Z value write-in operation ON, and drawing the resultant objects in the first frame buffer 11, and further drawing therein the objects to be α -blended after being processed under the conditions of α -blending ON, Z test ON and Z value write-in operation OFF.

[0065] The embodiment described in the above is only part of the examples of the present invention. It is therefore to be understood that the present invention may be practiced in any modifications depending on the design or the like otherwise than as specifically described herein without departing from the scope and the technical spirit thereof. For example, the number of frame image storage areas is by no means limited to two, and may be three or more. In the present invention, using a plurality of frame image storage areas to thereby distribute the processing will result in speeding up the image generation processing. In the present invention, three-dimensional α -blending is applicable not only to the foremost object, but also to the objects behind the foremost object having larger distances from the

